# Oracle (DBMS SQL) Lab Book :

1.1: Data Query Language:

```
1 select Staff_Name as name, Design_Code as code from staff_master where
Hiredate<STR_TO_DATE('Jan-2003','%M-%Y') and Staff_sal between 12000 and 25000;
```

```
2 select Staff_Code, Staff_Name, Dept_Code from staff_master where (year(CURDATE()) -
year(Hiredate)) >= 18 order by (year(CURDATE()) - year(Hiredate));
```

```
3 select * from staff_master where Mgr_code is null;
```

```
4 select * from book_master where (STR_TO_DATE(Book_pub_year,'%Y') between 2001 and 2004)
and Book_name like '%&%';
```

```
5 select Staff_name where Staff_Name like '%_%';
```

2.1: Single Row Functions:

```
1 SELECT Staff_Name,LPAD(staff_sal,15,'$') FROM staff_Master;
```

```
2 SELECT Student_Name,DATE_FORMAT(student_dob, "%M, %d %Y") from Student_Master where
WEEKDAY(student_dob) IN (5,6);
```

```
3 SELECT Staff_name, ROUND(TIMESTAMPDIFF(MONTH, HIREDATE, CURDATE()))AS
Months_Worked FROM Staff_Master ORDER BY HireDate;
```

```
4 select Staff_Name from Staff_Master Where DAYOFMONTH(hiredate)<16 and extract(Month from
hiredate)=12 ;
```

```
5 select Staff_Name,hiredate, staff_sal,
case
when staff_sal>=50000 then 'A'
when staff_sal >=25000 and staff_sal<5000 then 'B'
when staff_sal >=10000 and staff_sal<25000 then 'C'
else 'D'
end
from staff_master;
```

6 FUNCTION DOES NOT EXIST

```
7. SELECT LOCATE("i","Missisippi",3) AS i_position;
```

```
8. SELECT DATE_FORMAT(DATE(LAST_DAY(NOW()) - ((7 + WEEKDAY(LAST_DAY(NOW())) - 4) % 7)),
'%D of %M, %Y') AS PAY_DATE;
```

9. DECODE IS NOT AVAILABLE IN MYSQL.

## 2.2: Group Functions:

1. SELECT Dept_code,max(Staff_sal) AS Maximum,Min(Staff_Sal) AS Minimum,ROUND(Avg(Staff_sal)) AS Average FROM Staff_master GROUP BY Dept_code;

2. SELECT Deptno,COUNT(Deptno) AS Total_Number_of_Managers FROM emp WHERE JOB IN('MANAGER')GROUP BY Deptno,job;

3. SELECT Deptno,sum(sal) FROM emp WHERE JOB NOT IN('MANAGER') GROUP BY Deptno HAVING sum(sal) >2000 ORDER BY Deptno;

## 3.1: Joins and Subqueries:

1.SELECT S.Staff_Name, D.Dept_Code, D.Dept_Name, S.staff_sal FROM staff_master S, department_master D WHERE S.Dept_code=D.Dept_code AND Staff_sal >20000;

2. SELECT S.Staff_Code AS STAFF , S.Staff_Name AS STAFF, D.Dept_Name, S.Mgr_code AS MGRR FROM staff_master S,department_master D WHERE S.Dept_code=D.Dept_code;

3. SELECT S.Student_Code,S.Student_Name,B.Book_code,C.Book_name FROM student_master S,book_transactions B, book_master C WHERE S.Student_Code=B.Student_Code AND DATE_FORMAT(B.Book_expected_return_date,'%d-%b-%Y')LIKE DATE_FORMAT(SYSDATE(), '%Y-%m-%d');

4 SELECT staff_master.Staff_Code, staff_master.staff_name, department_master.Dept_Name, designation_master.Design_Name, book_master.Book_code, book_master.Book_name, book_transactions.Book_issue_Date FROM staff_master INNER JOIN department_master
ON staff_master.dept_code = department_master.dept_code INNER JOIN designation_master ON staff_master.design_code = designation_master.design_code INNER JOIN book_transactions ON staff_master.staff_code = book_transactions.staff_code INNER JOIN book_master ON book_transactions.book_code = book_master.book_code
WHERE book_transactions.Book_issue_Date >= CURRENT_DATE() - INTERVAL 30 DAY;

5 SELECT staff_master.Staff_Code, staff_master.Staff_Name, department_master.Dept_Name, designation_master.Design_Name, book_master.Book_code, book_master.Book_name, book_master.Book_pub_author, (5*(CURRENT_DATE() - book_transactions.Book_expected_return_date)) as fine FROM staff_master INNER JOIN department_master ON staff_master.dept_code = department_master.dept_code INNER JOIN designation_master ON staff_master.design_code = designation_master.design_code INNER JOIN book_transactions ON staff_master.staff_code = book_transactions.staff_code INNER JOIN book_master ON book_transactions.book_code = book_master.book_code WHERE book_transactions.Book_actual_return_date = null;

6 SELECT staff_master.Staff_Code, staff_master.Staff_Name, staff_master.Staff_sal FROM staff_master WHERE staff_master.Staff_sal < (select avg(staff_master.Staff_sal) from staff_master);

7

```
8 select s.staff_code,s.staff_name,d.dept_name from staff_master s INNER JOIN
department_master d ON s.dept_code=d.dept_code INNER JOIN book_transactions b ON
b.staff_code=s.staff_code;
```

```
9 select s.student_code,s.student_name,d.dept_name,count(dept_name) from student_master s
INNER JOIN department_master d ON s.dept_code=d.dept_code  group by dept_name;
```

10

```
11 SELECT (manager.ename),COUNT(*) from emp employee join emp manager on
employee.MGR=manager.EMPNO GROUP BY manager.EMPNO;
```

12

```
13 SELECT dept.dept_code,dept.dept_name,count(staff.staff_code) AS "employee_count" FROM
staff_master staff JOIN department_master dept ON staff.Dept_Code=dept.Dept_code GROUP BY
dept.Dept_code;
```

## 4.1: Database Objects:

```
1)create table customer (cust_id int(5),cust_name varchar(20),address1 varchar(30),address2
varchar(30));
```

```
2)alter table customer change cust_name custName varchar(30) not null;
```

```
3 a)alter table customer add column Gender  Varchar(1) ,add column Age  int(3),
add column PhoneNo int(10);
```

```
3 b)rename table customer to cust_table;
```

```
4)insert into cust_table values (1000, 'Allen', '#115 Chicago', '#115 Chicago', 'M', 25, 7878776);
insert into cust_table values (1001, 'George', '#116 France', '#116 France', 'M', '25', 434524);
insert into cust_table values (1002, 'Becker', '#114 New York', '#114 New York', 'M', '45', 431525);
```

```
5)alter table cust_table add constraint custId_Prim primary key (cust_id);
```

6) duplicate entry for '1002' key primary

```
7. alter table customer disable constraint CustId_Prim;
   insert into customer values (1002, 'Becker', '#114 New York', '#114 New york', 'M', '45', '431525');
    insert into customer values (1003, 'Nanapatekar', '#115 India', '#115 India', 'M', '45', '431525');
```

```
8. alter table customer enable constraint CustId_Prim;
```

```
9. alter table customer drop constraint CustId_Prim;
    insert into customer values (1002, 'Becker', '#114 New York', '#114 New york', 'M', '45', '431525');
```

```
   insert into customer values (1003, 'Nanapatekar', '#115 India', '#115 India', 'M', '45', '431525');
```

```
10. Truncate table customer;
```

```
11. Alter table customer add column E_mail varchar(30);
```

```
12. Alter table customer drop column E_mail;
```

```
13.CREATE TABLE IF NOT EXISTS `supplier` (

  `supplier_id` int(10) NOT NULL,
  `supplier_name` varchar(50) NOT NULL,
  `address1` VARCHAR(50) DEFAULT NULL,
  `address2` varchar(50) NOT NULL,
  `contact_no` INT(10) DEFAULT NULL,
  PRIMARY KEY (`supplier_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
14.DROP TABLE supplier;
CREATE TABLE IF NOT EXISTS `customer_master` (
  `customer_id` int(5) NOT NULL,
  `customer_name` VARCHAR(30) NOT NULL,
  `address1` VARCHAR(30) NOT NULL,
  `address2` VARCHAR(30),
  `gender` VARCHAR(1),
  `age` INT(3),
  `phone_no` INT(10),
  CONSTRAINT CustId_PK PRIMARY KEY (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
15. CREATE TABLE IF NOT EXISTS `accounts_master` (
  `customer_id` int(5) NOT NULL,
  `account_number` INT(10) NOT NULL,
  `account_type` VARCHAR(3) NOT NULL,
  `ledger_balance` INT(10),
  CONSTRAINT Acc_PK PRIMARY KEY (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```